

# 基于横向拓展窗口的快速图像中值滤波算法

张海青 杜晓晴 赵 釜 李小涛

(重庆大学光电工程学院光电技术及系统教育部重点实验室, 重庆 400044)

**摘要** 中值滤波是一种基于空域排序统计的图像降噪方法,可在滤除噪声的同时有效保护图像的细节信息。传统中值滤波一般采用冒泡排序法来对方形窗口内的像素灰度值进行排序操作,像素之间的比较次数多,处理时间长,不利于实时图像处理。基于图像并行处理的思想,提出了一种基于横向拓展窗口的快速中值滤波算法。算法将传统的方形窗口进行横向拓展,从而改变滤波窗口漫游的滑动步长,同时可以有效利用算法执行过程中的比较信息,达到提高整幅图像处理速度的目的。实验结果表明,算法能够满足图像降噪高精度、高实时性要求。所提出的算法通过牺牲芯片面积来换取处理速度,适宜于在现场可编程门阵列等硬件上实现。

**关键词** 图像处理; 图像降噪; 中值滤波; 拓展窗口

中图分类号 TP274.2 OCIS 100.2980 200.4960 文献标识码 A

## Fast Image Median Filtering Algorithm Based on Transverse Extended Window

Zhang Haiqing Du Xiaoqing Zhao Fu Li Xiaotao

(Key Laboratory of Optoelectronic Technology and Systems, Ministry of Education, College of Optoelectronic Engineering, Chongqing University, Chongqing 400044, China)

**Abstract** Median filtering is a spatial-order-statistics based image denoising method, which can preserve image details well while eliminating noise. In traditional median filters, the pixels in a square window is often sorted by bubble sorting, which demands large comparison times and long processing time, and is unfit for real-time image processing. Based on image parallel processing technology, a fast median filtering algorithm based on extended window is presented. In the proposed algorithm, the traditional square window is extended transversely, so the sliding step of window roaming can be increased by pixels to achieve much higher processing speed for the whole image, while the comparison results within the process of executing the algorithm are effectively used. The result shows that the algorithm can achieve high precision and real-time ability in image denoising. The algorithm improves the image processing speed by the cost of more area of hardware chip, and it is suitable for implementation on hardware such as field programmable gate array.

**Key words** image processing; image denoise; median filtering; extended window

### 1 引言

在数字图像的获取和传输过程中,传感器和传输通道经常会产生噪声。噪声的存在极大地降低了图像质量,对图像分割、特征提取、图像识别等后续处理将产生不可预料的影响,因此消除噪声在图像处理中具有非常重要的作用<sup>[1]</sup>。中值滤波是一种常用的图像去噪方法,基于排序统计理论进行非线性平滑滤波。其原理是首先确定一个以某一像素为中心的邻域,一般为方形邻域,然后对邻域中各个像素的灰度值进行

收稿日期: 2009-08-15; 收到修改稿日期: 2009-10-27

基金项目: 重庆大学“211工程”三期创新人才计划建设项目(S-09112)资助课题。

作者简介: 张海青(1983—),男,硕士研究生,主要从事图像处理方面的研究。E-mail: wavelet\_06215@163.com

导师简介: 杜晓晴(1978—),女,博士,副教授,主要从事光学探测技术方面的研究。

E-mail: duxq@cqu.edu.cn(通信联系人)

排序, 取其中间值作为中心像素的校正灰度值。中值滤波运算简单而且速度快, 在滤除叠加白噪声和长尾叠加噪声方面显示出极好的性能, 同时能够很好地保护图像的边缘、锐角等细节信息<sup>[2]</sup>。

窗口选择和排序实现是中值滤波算法的两个关键部分, 直接影响到算法的执行速度与滤波效果。传统中值滤波一般采用冒泡排序法, 通过遍历整个数列且若干重复操作实现, 包含大量重复比较过程, 处理时间长, 不利于实时图像处理。一般快速中值滤波则采取了并行处理的方法, 避免了大量比较操作, 大幅降低了算法的复杂度。对于  $3 \times 3$  的窗口数列, 只需 19 次比较, 相对于传统冒泡排序法的 36 次比较, 算法复杂度降低了近 50%<sup>[3]</sup>。并且这种并行处理算法适宜于在硬件上实现。本文结合这种图像并行处理的思想, 提出了一种基于横向拓展窗口的快速中值滤波算法。与一般快速中值滤波算法相比, 所提出的中值滤波算法在执行速度和实时性上都有很大提高, 并且适宜于在现场可编程门阵列等硬件上实现。

## 2 基于横向拓展窗口的快速中值滤波算法原理

一般快速中值滤波的处理步骤是: 第一步对每行进行冒泡排序, 得到各行的最大、最小和中值; 第二步比较得到 3 个最大值中的最小值、最小值中的最大值, 中值中的中值; 第三步对上一步得到的 3 个值进行冒泡排序, 得到其中值, 即为中值滤波输出<sup>[4]</sup>, 如图 1 所示。

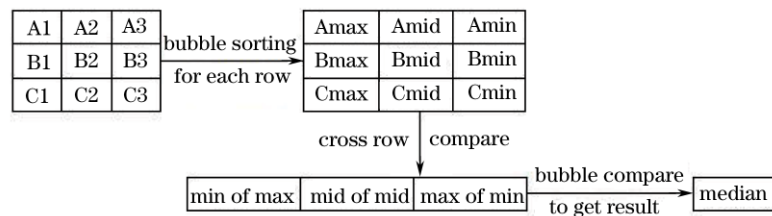


图 1 一般快速中值滤波算法流程  
Fig.1 Process of fast median filtering algorithm

虽然一般快速中值滤波较传统中值滤波有很大改进, 但是其相邻两点的处理共用 6 个像素点的比较信息, 占窗口像素总数的  $2/3$ , 这就意味着有接近一半的比较操作需要重复进行。由此可见, 通过降低相邻处理周期的相关度, 即共用比较信息占单个周期比较信息的比例, 能够达到充分利用比较信息、减少重复比较操作的目的, 从而提高图像处理速度。基于这种思想, 提出了一种基于横向拓展窗口的快速中值滤波算法, 下面以  $3 \times 4$  窗口为例来说明算法的实现原理。

首先对  $3 \times 4$  窗口内的像素按列进行编号, 如图 2 所示。处理时, 第一步对每列进行冒泡排序, 得到各列的最大、最小和中值, 如第一列的 A\_MAX、A\_MIN 和 A\_MID, 依此类推; 第二步对中间两列的第一步比较结果进行交叉比较, 得到 BC 两列最大值中的较小值 MIN\_OF\_MAX、最小值中的较大值 MAX\_OF\_MIN, 以及中值的比较结果 MAX\_OF\_MID 和 MIN\_OF\_MID; 第三步将 AD 两列的第一步比较结果分别与第二步比较结果进行交叉比较, 得到 ABC 三列最大值中的最小值 MIN\_OF\_MAX1, 最小值中的最大值 MAX\_OF\_MIN1 和中值中的中值 MID\_OF\_MID1, 同样还有后三列的 MIN\_OF\_MAX2, MAX\_OF\_MIN2 和 MID\_OF\_MID2。

A1	B1	C1	D1
A2	B2	C2	D2
A3	B3	C3	D3

图 2 对  $3 \times 4$  窗口内像素编号  
Fig.2 Arrangement of pixels within  $3 \times 4$  window

现在用排除法来求中值输出<sup>[5]</sup>。对于窗口内左边 9 个像素点, 最大值中的最大值即为 9 个点值的最大

值, 最小值中的最小值即为 9 个点值的最小值。最大值中的中值至少大于 5 个点值, 即本列的中值和最小值、以及最大值中的最小值和该列其他 2 个值, 同理最小值中的中值至少小于 5 个点值。中值中的最大值至少大于 5 个点值, 即本列最小值、其他两列的中值和最小值, 同理中值中的最小值至少小于 5 个点值。综上所述, 可能为中值输出的只有 MIN\_OF\_MAX1, MAX\_OF\_MIN1 和 MID\_OF\_MID1, 因此对这 3 个值进行比较得出其中值 MEDIAN1, 即为 9 个点值的中值。同理对于窗口右边 9 个像素点, 可以通过比较 MIN\_OF\_MAX2、MAX\_OF\_MIN2 和 MID\_OF\_MID2, 得出其中值 MEDIAN2。

故算法的第 4 步对第 3 步的两组结果分别进行冒泡比较, 并得出两个中值输出 MEDIAN1 和 MEDIAN2, 分别作为 B2 和 C2 点的灰度校正值。

显然, 新算法的相邻处理周期相关度降为 50%, 有效减少了重复比较操作的执行, 同时也大幅减少了比较次数。对于 3×4 窗口, 新算法的比较次数为 29 次, 相对于一般快速中值滤波的 38 次减少了近 30%, 如表 1 所示。比较次数的减少, 意味着算法复杂度的降低和图像滤波处理速度的提高。

表 1 3 种中值滤波方案的比较  
Table 1 Comparison of the three median filtering methods

Filtering method	Window	Period	Compare times	Move step /pixel
Bubble way	3×3	2	72	1
Normal fast way	3×3	2	38	1
Proposed way	3×4	1	29	2

新算法的一个重要改进体现在宽矩形窗口的选择。不同于既有算法的每个处理周期只校正一个像素点, 新算法每执行一次, 输出两个中值, 分别为中间两个像素的灰度校正值, 这使得窗口移动步长调节为 2 pixels。

### 3 算法验证及结果分析

通过以上研究分析, 作者在 MATLAB 环境下对新提出算法的可行性进行了验证, 并以峰值信噪比为评价标准, 对 3 种算法的模拟图像滤波效果做了对比分析。图 3(a)是叠加了 5% 椒盐噪声的原图, (b)~(d) 分别为 3 种中值滤波算法的处理结果图。可以看出, 新算法能够达到与前两种算法同等的视觉处理效果。表 2 中, 通过改变椒盐噪声的比例, 对比了 3 种算法的处理结果与加噪原图的峰值信噪比 (PSNR), 以及处理结果与标准原图的峰值信噪比 (PSNR')。



图 3 (a)加 5% 椒盐噪声的原图; (b) 冒泡法处理结果; (c) 一般快速中值滤波处理结果; (d) 新算法处理结果  
Fig.3 Original image with 5% impulse noise (a), result of bubbling method (b), fast median filtering (c) and the proposed method (d)

表 2 3 种算法处理的 PSNR 对比  
Table 2 PSNR comparison of the three filtering methods

Filtering method	Pepper & salt (1%)		Pepper & salt (5%)		Pepper & salt (10%)	
	PSNR /%	PSNR' /%	PSNR /%	PSNR' /%	PSNR /%	PSNR' /%
Bubble way	40.3634	41.2981	37.6913	40.8883	35.6839	40.3330
Normal fast way	39.4240	41.1539	37.1185	39.6974	35.2691	39.0897
Proposed way	39.1956	39.9033	36.9904	39.5019	35.1853	38.9165

从表 2 可以看出, 3 种算法的 PSNR 和 PSNR' 只有很少的误差, 说明新算法的去噪效果和相对于标准原图的保真度都能够达到传统中值滤波的水平。

#### 4 硬件仿真及结果分析

用硬件来实现图像处理算法有并行处理和流水线技术等独特的优势, 这使得其在实时图像处理中得到越来越广泛的应用<sup>[6]</sup>。中值滤波算法正是基于这一思想, 设计中在硬件开发平台上对新算法进行了仿真, 并与一般快速中值滤波做了对比。

本文在 QUARTUSII 开发平台上, 使用 Verilog HDL 语言实现滤波算法, 并利用 QUARTUSII 自带的仿真工具对滤波算法进行了功能仿真和时序仿真。图 4 为新算法的时序仿真波形图, 从图中可以看出, 自第一组有效值开始, 经过中值滤波后, 输出值即为窗口内像素值的中间值, 有效验证了其逻辑上的正确性。

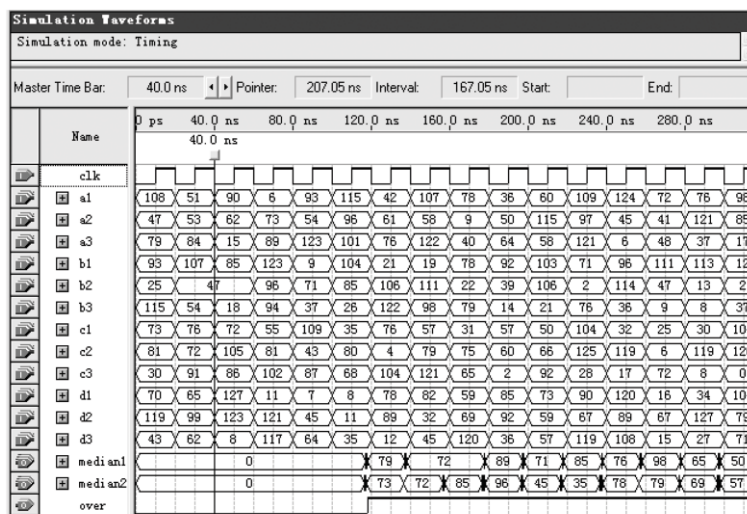


图 4 中值滤波的仿真结果

Fig.4 Simulation result of median filter module

实验过程中还对两种算法的硬件资源消耗和处理速度做了比较, 发现新算法通过相对多占用 49% 的芯片资源, 将图像滤波速度提高了近 1 倍, 这具有重要的理论和现实意义。图 5 是应用新算法, 在以 Altera 公司的 Cyclone II EP2C35F672C6 FPGA 为核心的硬件平台上, 对加 5% 椒盐噪声图像的处理结果。

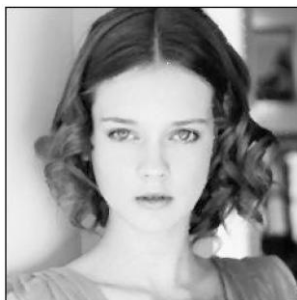


图 5 中值滤波的硬件仿真结果

Fig.5 Hardware simulation result of median filtering

#### 5 结 论

图像的预处理包括很多算法, 如平滑去噪处理、锐化等。为了有效地去除噪声, 应根据噪声类型采用不同的滤波方法<sup>[7]</sup>。本文针对椒盐噪声的滤除, 分析研究了中值滤波算法的原理和特点, 并提出了基于横向拓展窗口的快速并行中值滤波算法及其 FPGA 硬件实现方案。算法通过改变窗口形状, 对待处理图像像

素采取批处理的方式, 每个周期处理两个以上像素点, 充分利用了丰富的硬件资源和 FPGA 的并行流水线处理技术, 有效地提高了图像的中值滤波速度, 这在实时图像处理方面有很高的应用价值。后续研究将集中探讨拓展窗口在其他算法上的应用以及实时图像处理系统的实现。

### 参 考 文 献

- 1 Tang Quanhua, Zhou Yan, Jin Weidong. Research and improvement to median filter algorithm[J]. *Control & Automation*, 2008, **24**(8): 187~189  
唐权华, 周 艳, 金炜东. 中值滤波算法的研究与改进[J]. 微计算机信息, 2008, **24**(8): 187~189
- 2 Zhao Fu. Research on Image Capture and Preprocessing System Based on Cyclone II Series FPGA[D]. Chongqing: Chongqing University, 2009. 57~68  
赵 釜. 基于Cyclone II系列FPGA的图像实时采集与预处理系统[D]. 重庆: 重庆大学, 2009. 57~68
- 3 Wang Yuxin, He Yuanyuan, Guo He *et al.*. FPGA based algorithm of fast median filter[J]. *App. Res. Comput.*, 2009, **26**(1): 224~226  
王宇新, 贺圆圆, 郭 禾 等. 基于FPGA的快速中值滤波算法[J]. 计算机应用研究, 2009, **26**(1): 224~226
- 4 Hu Yueli, Ji Huijie, Wu Pin *et al.*. Research on image filtering algorithm and its FPGA implementation[J]. *Comput. Meas. Control*, 2008, **16**(11): 1672~1675  
胡越黎, 计慧杰, 吴 频 等. 图像的中值滤波算法及其FPGA实现[J]. 计算机测量与控制, 2008, **16**(11): 1672~1675
- 5 Wan Haijun, He Dongjian, Xu Shangzhong. The realization of image median filter algorithm based on FPGA[J]. *Control & Automation*, 2008, **24**(7): 280~282  
万海军, 何东健, 徐尚中. 基于FPGA的图像中值滤波算法硬件实现[J]. 微计算机信息, 2008, **24**(7): 280~282
- 6 Stephen Brown, Zvonko Vranesic. Fundamentals of Digital Logic with Verilog Design[M]. Beijing: China Machine Press, 2007. 405~433  
斯蒂芬, 兹翁科. 数字逻辑基础与Verilog设计[M]. 北京: 机械工业出版社, 2007. 405~433
- 7 Wang Ailing, Ye Mingsheng, Deng Qiuxiang. MATLAB R2007 Image Processing Technology and Application[M]. Beijing: Publishing House of Electronics Industry, 2008. 49~53  
王爱玲, 叶明生, 邓秋香. MATLAB R2007图像处理技术与应用[M]. 北京: 电子工业出版社, 2008. 49~53